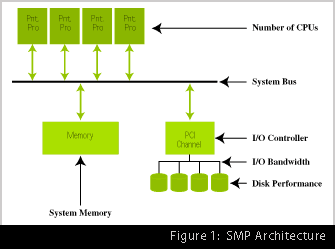
**Non-Uniform memory Access**

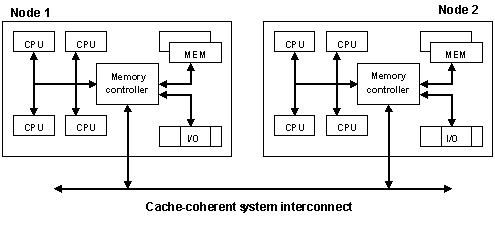
1. **Symmetric Multi Processor (SMP)**

* Trước tiên xem xét phương thức tiếp cận bộ nhớ theo phương thức truyền thống là SMP (Bộ đa xử lý đối xứng).
* Với phương thức SMP, tất cả các bộ nhớ truy cập sẽ được gửi tới cùng một vùng nhớ chia sẻ, mỗi bộ xử lý có quyền truy cập bộ nhớ và sử dụng số lượng I/O giống nhau.
* Hệ thống có thể hoạt động tốt với số lương nhỏ CPUs.
* Khi có nhiều tài nguyên cần tính toán cùng một lúc, lúc này các CPUs phải tranh giành quyền truy cập bộ nhớ và sử dụng I/O.
* Phương pháp này không tốt đối với trường hợp xử lý nhiều tài nguyên cùng lúc

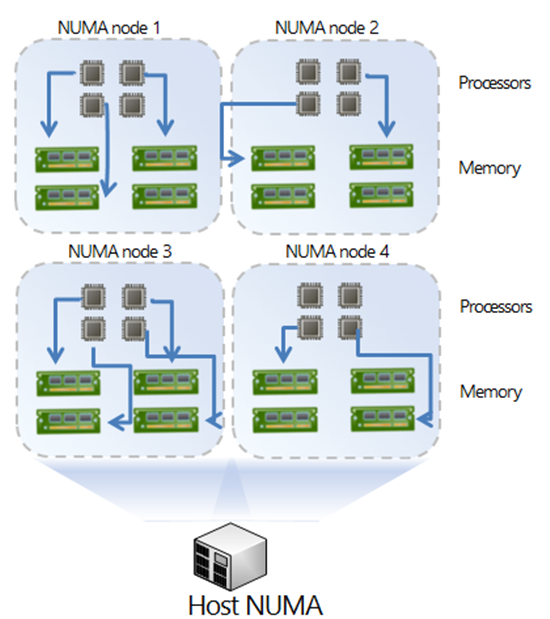


1. **Non-Uniform memory Access (NUMA)**

NUMA - Bộ đa xử lý truy cập bất đối xứng

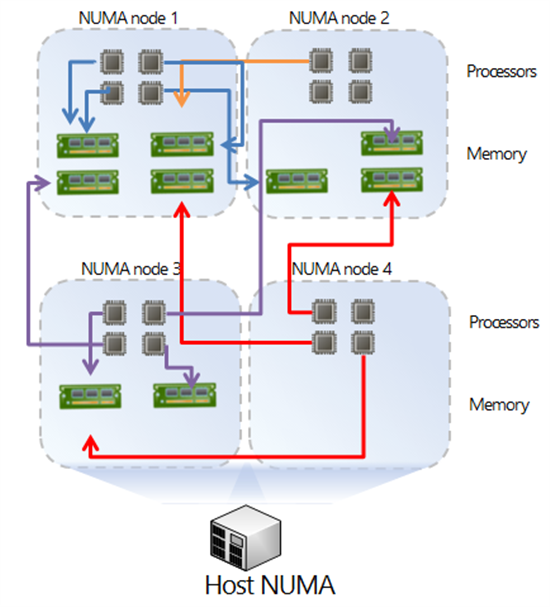


* NUMA được ra đời để giải quyết cho vấn đề thắt cổ chai bằng cách nhóm các tài nguyên tính toán và bộ nhớ thành các node( nốt).
* Mỗi một node sẽ được kết nối với nhau qua các cache-coherent bus.
* Memory và CPU trong cùng một node NUMA đang chạy các chương trình xử lý được coi như là vùng bộ nhớ nội bộ và các vùng nhớ đang chạy các chương trình xử lý khác không nằm trong cùng một node được gọi là remote.
* Với công nghệ Non-uniform memory access thì bộ nhớ sẽ được truy cập nhanh hơn, trong khi các bộ vi xử lý sẽ truy cập vào vùng nhớ nội bộ của chính nó thay vì các vùng nhớ remote.
* *Ví dụ về Optimal NUMA Configuration*



Các node NUMA được cấu hình tối ưu:

* Mỗi node NUMA có cùng một số lượng bộ nhớ giống nhau.
* CPU và Memory được phân bổ trên cùng một node NUMA.
* *Ví dụ về Non-Optimal NUMA Configuration*



* Hệ thống NUMA ở đây bị mất cân bằng. Mỗi node NUMA có số lương Memory khác nhau.
* Node NUMA thứ hai có số lượng Memory lẻ vì thế sẽ ảnh hưởng nếu hệ thống sử dụng bộ nhớ đan xen ( ngoài ra còn phụ thuộc vào các yếu cầu trên bo mạch chủ đan xen như: pairs, trior, etc…).
* Node NUMA thứ hai và thứ ba không có đủ bộ nhớ nội bộ, điều này dẫn tới một số vùng nhớ sẽ truy cập ở nội bộ và một số khác sẽ dùng các bộ nhớ remote. Điều này có thể dẫn đến có khả năng truy cập các vùng nhớ không phù hợp và làm giảm hiệu suất.
* Node Numa thứ tư không có bộ nhớ cục bộ do đó tất cả các truy cập bộ nhớ sẽ là remote.

1. **SQLServer và NUMA**

* NUMA có thể kết hợp bộ nhớ với CPUs thông qua Hardware NUMA hoặc bằng cách cấu hình bộ nhớ SQL Server (Soft NUMA).
* Mỗi node NUMA (Hardware NUMA hay Soft NUMA) sẽ luôn có một cổng liên kết I/O completion dùng để quản lý mạng I/O.
* Việc này sẽ giúp các I /O được xử lý thông qua nhiều cổng.
* Khi việc kết nối từ client được thực hiện tới SQL Server, nó sẽ được gom lại thành một nodes.
* Tất cả các yêu cầu từ client sẽ được xử lý tại chính node đó.
* Mỗi instant của SQL Server được xử lý trong một môi trường NUMA
* Một bản ghi SQL Server ghi lại thông tin mô tả cấu hình NUMA cho mỗi node cùng với CPU mask.
* *Truy xuất bảng:*

***sys.dm\_os\_memory\_nodes*** *và* ***sys.dm\_os\_memory\_node\_access\_stats***

*để lấy thông tin về các node bộ nhớ NUMA vật lý và số liệu thống kê node truy cập được nhóm theo loại trang.*

1. **Hardware NUMA**

* Hardware NUMA đã được đưa vào trong Windows để nhóm các CPUs lại với nhau.
* Ví dụ: với 16-way box chúng ta có thể có 4 nodes NUMA, mỗi node sẽ có 4 CPUs. Điều này cho phép chúng ta có nhiều bộ nhớ nội bộ hơn cho việc thiết lập tác vụ cho các chương trình xử lý.
* Khi một thread đang chạy trên một node Hardware NUMA, trình quản lý bộ nhớ của SQL Server sẽ cố gắng xử lý việc cấp phát bộ nhớ từ bộ nhớ liên kết với các node NUMA để xác định các vùng lưu trữ. Các bộ nhớ đệm sẽ được phân phối dự trên các node Hardware NUMA
* Hiệu quả cho việc Thread truy cập vào bộ nhớ từ vùng nhớ đệm nằm trên vùng nhớ nội bộ thay vì truy cập từ các vùng nhớ ngoài.
* *Thực hiện truy vấn sau để tìm số node bộ nhớ có sẵn trong SQL Server*

**SELECT DISTINCT memory\_node\_id**

**FROM sys.dm\_os\_memory\_clerks**

* Nếu giá trị trả về chỉ một node bộ nhớ duy nhất là node 0 thì Hardware NUMA không có hoặc phần cứng được cấu hình đan xen (non-NUMA).

1. **Soft NUMA**

* SQL Server cho phép nhóm CPUs thành các node hoặc chia nhỏ các node Hardware NUMA thành các node nhỏ hơn.
* Soft NUMA giúp giảm I/O và nút cổ chai trên các máy tính với nhiều CPUs và không Hardware NUMA.
* Với mỗi node NUMA có một I/O thread và một lazy writer thread.
* Một số cấu hình phần cứng chia sẻ các nguồn tài nguyên chung như một L3/L4 cache. Bộ vi xử lý có thể được nhóm lại xung quanh các tài nguyên được chia sẻ để tạo ra các node Soft NUMA.
* Tham khảo thêm cách cấu hình SQL Server để sử dụng Soft NUMA

<http://msdn.microsoft.com/en-us/library/ms345357(v=sql.105).aspx>